

به نام خدا

نام توسعه دهنده :

میثم عقیلی 970088600

عنوان پروژه :

password storage website

توضیحات پروژه :

پروژه یک وبسایت برای ذخیره پسوردهای مختلف می باشد به طوری که در مواردی که نیاز است با رجوع به سایت بتوان به آنها دسترسی داشت.

اهداف پروژه :

- ذخیره اطلاعات پسوردها
- دسترسی راحت و ایمن به اطلاعات پسوردها

تکنولوژی های استفاده شده :

- Frontend : Html, css
- Backend : Python Django
- Database : sqlite

ساختار پروژه :

در پوشه `\src\user\templates\user` فایل های html مربوط به صفحات مختلف وجود دارد.

در پوشه `src\user\static` فایل های image و css و js وجود دارد.

در پوشه `src\user` فایل `views.py` مربوط به بخش `controller` و `backend functions` می باشد. فایل

`model.py` مربوط به دیتابیس و جداول و `urls.py` برای مسیریابی صفحات می باشد.

نحوه اجرای پروژه :

1-نصب کتابخانه های مورد نیاز `pip install -r requirements.txt`

2- اجرا `python manage.py runserver`

```
C:\Windows\System32\cmd.exe - python manage.py runserver
Microsoft Windows [Version 10.0.19045.3086]
(c) Microsoft Corporation. All rights reserved.

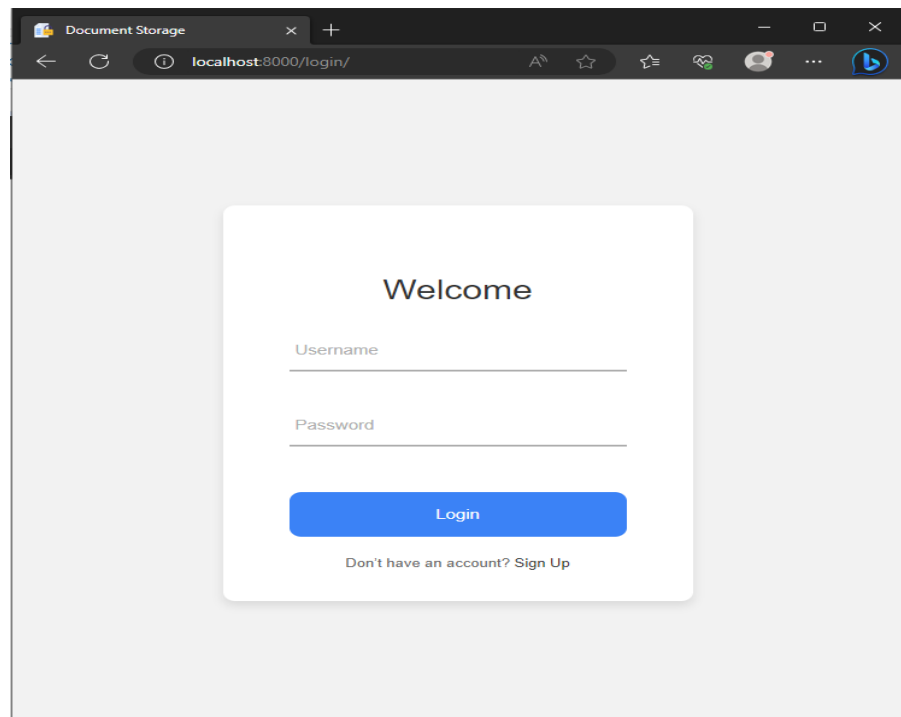
C:\Users\meysam\Desktop\github\password_storage\src>.venv\Scripts\activate.bat
(.venv) C:\Users\meysam\Desktop\github\password_storage\src>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 19, 2023 - 16:04:20
Django version 4.2, using settings 'project.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

-
```

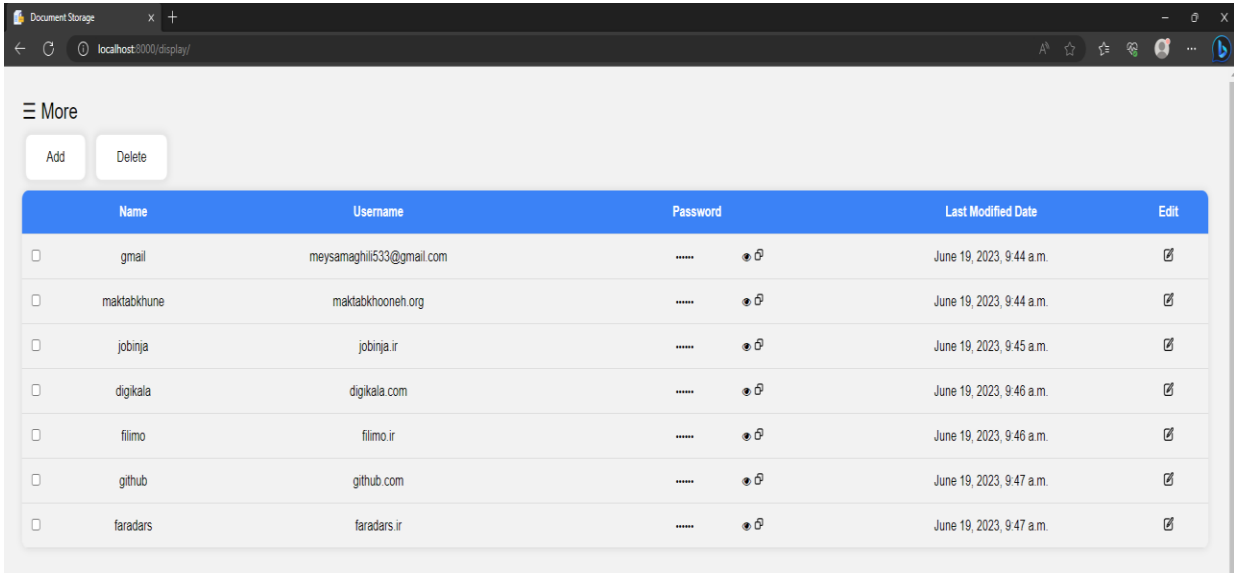
سایت بر روی ip localhost و port 8000 اجرا می شود.

در ابتدا صفحه login را داریم:



در صورت نیاز می توان ثبت نام نیز انجام داد.

سپس بعد از ورود به صفحه display وارد می شوید.



The screenshot shows a web browser window with the address bar displaying 'localhost:8000/display/'. The page has a header with a 'More' menu and 'Add' and 'Delete' buttons. Below this is a table with the following columns: Name, Username, Password, Last Modified Date, and Edit. The table contains seven rows of user data.

	Name	Username	Password	Last Modified Date	Edit
<input type="checkbox"/>	gmail	meysamaghi533@gmail.com	June 19, 2023, 9:44 a.m.	
<input type="checkbox"/>	maktabkhune	maktabkhooneh.org	June 19, 2023, 9:44 a.m.	
<input type="checkbox"/>	jobinja	jobinja.ir	June 19, 2023, 9:45 a.m.	
<input type="checkbox"/>	digikala	digikala.com	June 19, 2023, 9:46 a.m.	
<input type="checkbox"/>	filimo	filimo.ir	June 19, 2023, 9:46 a.m.	
<input type="checkbox"/>	github	github.com	June 19, 2023, 9:47 a.m.	
<input type="checkbox"/>	faradars	faradars.ir	June 19, 2023, 9:47 a.m.	

در این صفحه name , username , password های خود را مشاهده می کنید و امکان اضافه کردن و حذف و تغییر نیز وجود دارد. همچنین تاریخ آخرین تغییر و امکان کپی کردن پسورد نیز وجود دارد.

با کلیک بر روی More می توانید به پنل Logout دسترسی یابید.

توابع Views.py :

تابع register : با متد پست اطلاعات فرم را ثبت نام می کند تا کاربر جدید ایجاد شود.

```
def register(request):
    form = UserRegisterForm(request.POST or None)
    if request.method == 'POST':
        if form.is_valid():
            form.save()
            u = user(username=form.cleaned_data['username'], password=form.cleaned_data['password1'])
            u.save()
            messages.success(request, 'Successfully Registered.')
            return _login(request, form.cleaned_data['username'], form.cleaned_data['password1'])
    return render(request, 'user/register.html', {'form': form})
```

ورودی request و خروجی render display می باشد.

Login : امکان ورود کاربر از طریق ارتباط با دیتابیس را فراهم میکند.

```
def Login(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        return _login(request, username, password)
    form = AuthenticationForm()
    return render(request, 'user/login.html', {'form':form})

def _login(request, username, password):
    user = authenticate(request, username = username, password = password)
    if user is not None:
        login(request, user)
        return redirect('display')
    else:
        messages.error(request, f'Username or Password is incorrect.')
        form = AuthenticationForm()
        return render(request, 'user/login.html', {'form':form})
```

ورودی request و خروجی render display می باشد.

Display : اطلاعات پسورد ها را از دیتابیس گرفته و نمایش می دهد.

```
@login_required(login_url="login")
def display(request):
    u = user.objects.get(username=request.user.username)
    data = pass_data.objects.all().filter(user_id=u)
    return render(request, 'user/display.html', {'data':data})
```

صفحه display را با data نشان می دهد.

Delete_pass : پسورد را حذف می کند.

```
@login_required(login_url="login")
def delete_pass(request):
    form = delete_pass_form(request.POST or None)
    if request.method == "POST":
        if form.is_valid():
            names = literal_eval(form.cleaned_data['names'])
            names = [name[:-len('_check')] for name in names]
            u = user.objects.get(username=request.user.username)
            c1 = Q(user_id=u)
            c2 = Q(name__in=names)
            pass_data.objects.filter(c1 & c2).delete()
            messages.success(request, 'Successfully Removed.')
    return redirect('display')
```

با اتصال به دیتابیس رکورد موردنظر را حذف می کند.

Edit_pass : پسورد را ویرایش می کند.

```
@login_required(login_url="login")
def edit_pass(request, id):
    if request.method == "POST":
        form = edit_pass_form(request.POST)
        if form.is_valid():
            u = user.objects.get(username=request.user.username)
            c1 = Q(user_id=u)
            c2 = Q(id=id)
            p = pass_data.objects.get(c1 & c2)
            p.name = form.cleaned_data['name']
            p.username = form.cleaned_data['username']
            p.password = form.cleaned_data['password']
            p.last_modified_date = datetime.now()
            p.save()
            messages.success(request, 'Successfully Edited.')
            return redirect('display')
    if request.method == "GET":
        u = user.objects.get(username=request.user.username)
        c1 = Q(user_id=u)
        c2 = Q(id=id)
        p = pass_data.objects.get(c1 & c2)
        initial={'name':p.name,'username':p.username,'password':p.password}
        return render(request, 'user/edit_pass.html', {'initial':initial, 'id':id})
    else:
        return redirect('display')
```

Add_pass : پسورد را اضافه می کند.

```
@login_required(login_url="/login/")
def add_pass(request):
    form = add_pass_form(request.POST or None)
    if request.method == "POST":
        if form.is_valid():
            u = user.objects.get(username=request.user.username)
            p = pass_data(user_id=u, name=form.cleaned_data['name'], username=form.cleaned_data['username'], password=form.cleaned_data['password'])
            p.save()
            messages.success(request, 'Successfully Added.')
            return redirect('display')
    return render(request, 'user/add_pass.html', {'form':form})
```

ورودی request و با اتصال به دیتابیس پسورد را درج و صفحه display را render می کند.